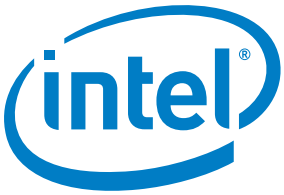


Intel Atom® Processor E3800 Product Family Board Support Package (BSP) for Windows* 10 IoT Core 32-Bit and 64-Bit Platforms

User Guide

January 2018

Revision 003

Revision Number: 003US

Contents

Figures

Tables

Revision History

Date	Revision	Description
January 2018	003	Intel Atom® Processor E3800 Product Family Board Support Package for Windows* 10 IoT Core 32-bit and 64-bit Platforms—MR1 Release
March 2016	002	Updated Section 2.0 Section 3.1: Revised Paragraph 1 Step 3. Revised adding subparagraphs A and B Inserted a new Step 4 Section 3.2.1: Added Note below Step 1 Section 3.4: Revised Step 2 environmental variable code Revised Step 3 Note
December 2015	001	Intel Atom® Processor E3800 Product Family Board Support Package for Windows* 10 IoT Core 32-bit and 64-bit Platforms—Gold Release

§

1.0 Introduction

1.1 Scope of Document

This document describes the installation of the Microsoft* Windows* OS 10 developer tools to create an image for the Microsoft* Windows* 10 IoT Core 32-bit and 64-bit operating systems, booting up the IoT Core image, and best known methods for platforms and software drivers.

This User Guide is intended for Original Equipment Manufacturers (OEMs) and Original Design Manufacturers (ODMs) who enable IoT Core drivers for the Windows* 10 IoT Core operating system with Intel Atom® E3800 processor, Intel Celeron® processor N2XXX, and Intel Celeron® processor J1XXX.

1.2 System Requirements

The operating systems supported are the Microsoft Windows 10 IoT Core 32-bit and 64-bit operating systems.

1.3 Acronyms and Terminology

Table 1. Terminology

Term	Description
ADK	Assessment and Deployment Kit
BSP	Board Support Package
COM Port	Communication Port
CRB	Customer Reference Board
DMA	Direct Memory Access
EHCI	Enhanced Host Controller Interface
FFU	Full Flash Update
GPIO	General Purpose Input/Output
HSUART	High Speed Universal Asynchronous Receiver/Transmitter
I2C*	Inter-Integrated Circuit*
NIC	Network Interface Card
ODM	Original Design Manufacturer

Term	Description
OEM	Original Equipment Manufacturer
PIO Mode	Programmed I/O Mode
RTM	Release to Manufacturing
SDK	Software Development Kit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
WDK	Windows Driver Kit

1.4 Reference Documents

Table 2. Reference Documents and Resources

Document	Document No./ Resource Location
MSDN* Subscriptions Website: Windows* 10 IoT Core Packages.msi	https://msdn.microsoft.com/en-us/subscriptions
Microsoft* IoT Core Downloads and Tools: IoT Dashboard	https://developer.microsoft.com/en-us/windows/iot/downloads
Windows* 10 Assessment and Deployment Kit (Windows ADK)	https://developer.microsoft.com/en-us/windows/hardware/windows-assessment-deployment-kit
GitHub* site with Windows* 10 IoT Core ADK add-ons	https://github.com/ms-iot/iot-adk-addonkit/releases/tag/v2.0
Microsoft IoT Core Manufacturing Guide	https://msdn.microsoft.com/en-us/windows/hardware/commercialize/manufacture/iot/iot-core-manufacturing-guide
Microsoft TechNet Library: WinPE: Create USB Bootable Drive	https://technet.microsoft.com/enus/library/hh825109.aspx
Future Technology Devices International (FTDI) Chip* Drivers	www.ftdichip.com/FTDrivers.htm
Intel Atom® Processor E3800 Product Family Datasheet	538136

§

2.0 Best Known Configuration

Table 3 shows the software tools required to create and deploy the Windows* 10 IoT Core operating system (32-bit and 64-bit Windows* 10 IoT Core) with custom drivers on the hardware.

Table 3. Best Known Configuration (BKC) for 32-bit and 64-bit Windows* 10 RS1 Build

Development Tools	Version
Operating System	Windows* 10 RS1 (Build 14393 or later) x86 or x64
Windows* 10 IoT Core Packages	10.0.14393
Windows* 10 IoT Core Dashboard	1.0.1711.14001
Windows* IoT Core ADK add-ons	10.0.14393
Windows* Assessment and Deployment Kit (ADK)	10.0.14393

§

3.0 Build Windows* 10 IoT Core Image

This section describes how to install Windows* 10 developer tools to create an image for the Windows 10 IoT Core 32-bit and 64-bit operating systems. It also describes booting up the IoT Core image on a Bayley Bay customer reference board (CRB) and upgrading I/O drivers (GPIO, I2C*, SPI, HSUART, and graphics) for the IoT Core image.

3.1 Install Windows* 10 Developer Tools

Use the following steps to install the Windows* 10 developer tools:

1. Install the Windows* 10 operating system on a development machine.
2. Build a custom Full Flash Update (FFU) image. To do this, download the following package from the MSDN* Subscriptions website (refer to Table 2) and install it to extract the FFU content *MSPackages* to the development machine:

   `Windows_10_IoT_Core_Packages.msi`

3. Download and install the IoT Core Dashboard (refer to Table 2).
4. Download and install the Windows* 10 ADK (refer to Table 2).

   In addition to the new and improved deployment tools used to automate a large-scale Windows* 10 deployment, the ADK includes the Windows* Assessment Toolkit and Windows* Performance Toolkit, which can assess the quality and performance of systems and components.
5. Download and install the Windows* 10 IoT Core ADK add-ons from the GitHub* page listed in Table 2. To do this, click **Clone or Download** > **Download ZIP**, and extract it to a folder, such as `C:\iot-adk-addonkit-2.0`.

3.2 Install the Board Support Package (BSP)

Use the following steps to install the Board Support Package:

1. Unzip the Intel Atom®/Celeron®/Pentium® Windows* 10 package IoT Core BSP, and copy the driver files to the ADK add-ons kit extracted folder, such as `C:\iot-adk-addonkit-2.0`. For example:
   - For 32-bit, copy the BYTx86 folder to C:\iot-adk-addonkit-2.0\Source-x86\BSP
   - For 64-bit, copy the BYTx64 folder to C:\iot-adk-addonkit-2.0\Source-x64\BSP
2. Create the IoT Core FFU image (refer to Section 3.3).

3.3 Create the IoT Core Full Flash Update (FFU) Image

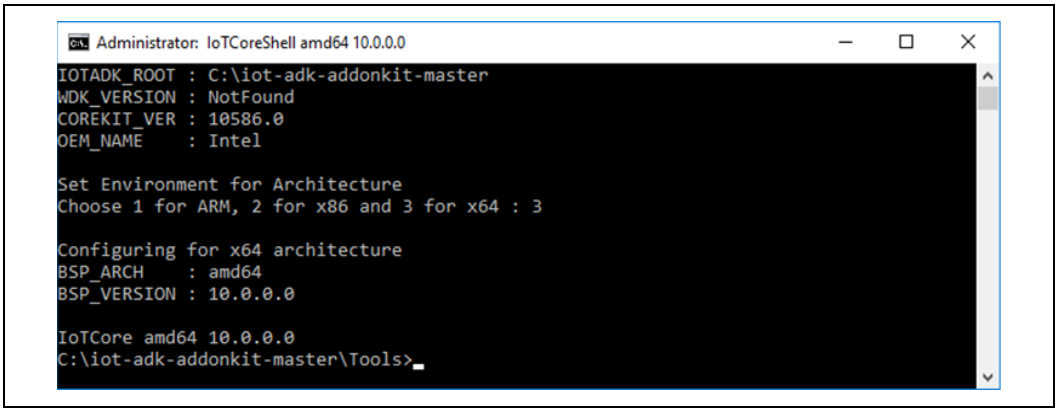
This section describes the steps to create an IoT Core FFU image.

1. Edit `C:\iot-adk-addonkit-2.0\Tools\setOEM.cmd` to change the OEM name. For example:

   ```
   >set OEM_NAME=Intel
   ```

2. Open an elevated Command Prompt, and run `C:\iot-adk-addonkit-2.0\IoTCoreShell.cmd`.
3. Select **3** to set **BSP_ARCH** as the x64 version, or select **2** for the x86 version. This opens a new IoTCoreShell command prompt window, as shown in Figure 1.

Figure 1. IoTCoreShell Environment Selection Screen



4. From the new IoTCoreShell command prompt, run the following command to install OEM test certificates:

   ```
   >"%WPDKCONTENTROOT%\tools\bin\i386\InstallOEMCerts.cmd"
   ```

Note: This command is only needed when building the package for the first time.

5. Run the following command to generate all cabinet files:

   ```
   >buildpkg all
   ```

6. Run the following command to create a project:

   ```
   >newproduct <Product_Name> <BSP_Name>
   ```

Note: Replace the *<Product_Name>* and *<BSP_Name>*. The *<BSP_Name>* for a 32-bit image is BYTx86, and for 64-bit image, it is BYTx64.

7. Run the following command to create a test FFU image:

   ```
   >buildimage <Product_Name> Test
   ```

Note: Ensure all removable storage drives (including micro SD and USB flash drives) are unplugged. The build will stall if any removable storage drives are plugged in.

8. The build process takes about 20–30 minutes. The `Flash.FFU` image can be found in the `C:\iot-adk-addonkit-2.0\Build\<CPU Type>\<Product_Name>\Test` folder.

Note: For more details, refer to the Microsoft IoT Core Manufacturing Guide (listed in Table 2).

3.4 Flash the Windows* 10 IoT Core FFU to a Storage Device

This section describes how to flash the IoT Core image to different storage devices, including:

- USB storage device
- SATA disk
- eMMC*

3.4.1 USB Storage Device

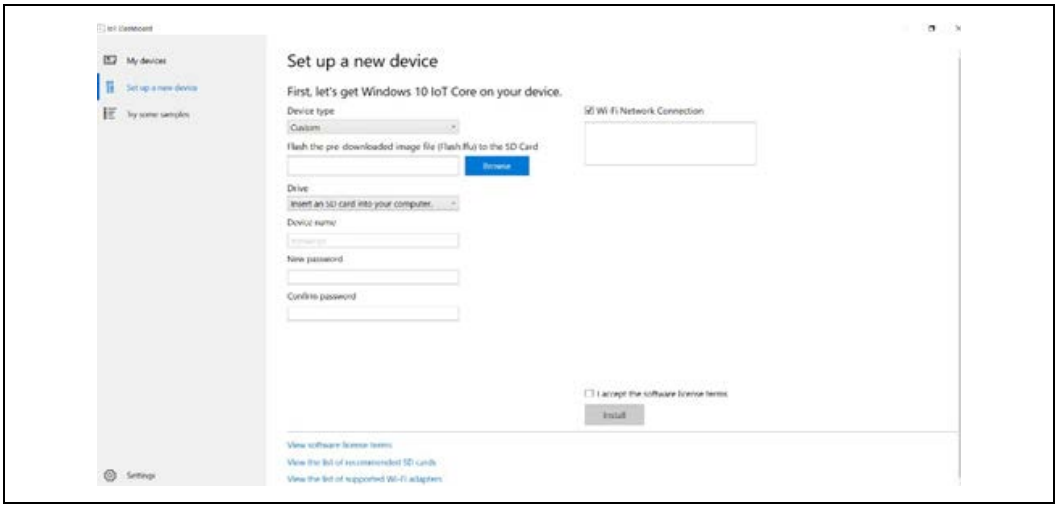
To flash the IoT Core image to a USB storage device:

1. Connect a USB storage device to the development machine. The storage device must have at least 16 GB of available storage space.
2. Download and install Windows* 10 IoT Core Dashboard (link provided in Table 2). Figure 2 shows the IoT Core Dashboard view.

Figure 2. IoT Core Dashboard Interface

3. Select **Set up a new device**, as shown in Figure 3. For **Device type**, choose **Custom**, and browse to the FFU image location.

Figure 3. Set Up a New Device Options



4. Select the drive to flash the FFU image (the USB storage device).
5. Once all necessary options are selected, click **Install** to start flashing the FFU image to the selected drive.

3.4.2 SATA Disk

To flash the IoT core image to a SATA disk:

1. Connect the SATA disk (with at least 16 GB of available storage space) to the development machine.
2. Open a command prompt, and log in as the administrator.
3. Run the `diskpart` command, run `list disk`, and find the number **N** of the SATA disk.
4. Run the `exit` command to quit the DiskPart terminal.
5. Run the following command, also shown in Figure 4, to flash the FFU image to the SATA disk:

```
C:\Program Files (x86)\Windows Kits\10\Assessment and
Deployment Kit\Deployment Tools\x86\DISM\dism.exe /Apply-Image
/ImageFile:C:\FFU\IoTCore.ffu /ApplyDrive:\\.\PhysicalDriveN
/SkipPlatformCheck
```

Figure 4. Deploy IoTCore.ffu into SATA HDD Using DISM

3.4.3 eMMC*

This section describes how to flash the image to an eMMC* using WinPE*. Ensure the system has the Windows* ADK installed with the Deployment Tools and Windows* Pre-installation Environment features.

1. Start the Deployment and Imaging Tools Environment as the administrator.
2. Create a working copy of the Windows PE* files. Specify x86 for 32-bit or amd64 for 64-bit, as follows:

   ```
   Copype amd64 C:\WinPE_amd64
   ```

3. Install Windows PE* to the USB flash drive as follows, assuming the USB flash device is the **F:** drive:

   ```
   MakeWinPEMedia /UFD C:\WinPE_amd64 F:
   ```

Note: For more information, refer to Microsoft TechNet Library article, WinPE: Create USB Bootable Drive (listed in Table 2).

4. Copy the FFU image to the root of an MBR partition WinPE disk.
5. Insert the USB flash drive into the Apollo Lake CRB.
6. Boot the CRB, and press **F2** to access the BIOS settings.
7. Navigate to **Boot Manager**, and select the USB device. It will boot Windows PE.
8. Run the following DISM command to flash the image to the eMMC* device:

   ```
   dism /apply-image /imagefile:d:\flash.ffu
   /applydrive:\\.\PhysicalDrive0 /skipplatformcheck
   ```

Note: The steps assume the WinPE USB is in the *D:*\ drive and eMMC is PhysicalDrive0. Use the `diskpart` utility to check the device drive letter, if necessary.

9. After the DISM completes, restart the platform, and select **EFI Hard Drive** on the Boot Option menu.

3.5 Boot Windows* 10 IoT Core on Bayley Bay CRB

This section describes how to start the Windows*IoT core on Bayley Bay CRB.

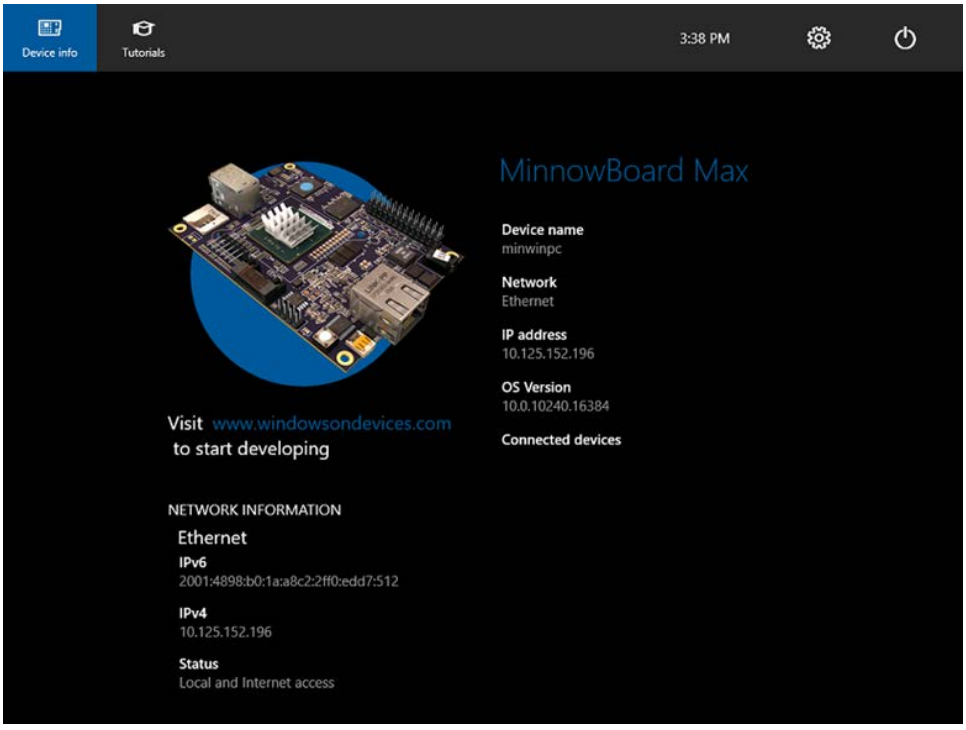
1. When the Bayley Bay CRB boots, press **F2** to access the BIOS settings.
2. In the BIOS settings, go to Device Manager, and choose **System Setup** and apply the following changes:
   - **Boot > OS Selection:** Select **Windows 8.X**.
   - **South Cluster Configuration > LPSS & SCC Configuration > LPSS & SCC Devices Mode:** Select **ACPI Mode**.
   - **South Cluster Configuration > LPSS & SCC Configuration > ACPI Reporting MMC/SD Media As:** Select **Non-Removable**.

3. Press **F4** to save the changes, or choose **Commit Changes and Exit**.
4. Navigate to the first level of the BIOS settings, and select **Boot Maintenance Manager**, choose **Boot Options**, and then select **Add Boot Option**.
5. In File Explorer, select **EFIESP** and press **Enter**
6. Navigate to **`bootia32.efi`** (located at …`EFI/Boot/bootia32.efi`), and press **Enter**.
7. Input the description, type, any name, such as *IoTCoreImage*.
8. Choose **Commit Changes and Exit**.
9. Navigate to the Change Boot Order, highlight the boot order list, and press **Enter**.
10. Select **IoTCoreImage,** and press **+** to move it to the top of the list.
11. Chose **Commit Changes and Exit**.

The Bayley Bay CRB should automatically boot to IoTCoreImage, as shown in Figure 5.

Note: After selecting the language, the device will boot. If a language is not selected, the device may restart. After the device is booted, the default application from the USB drive will launch and display the IP address of the Bayley Bay CRB.

Figure 5. Bayley Bay CRB IP Address Display



§

4.0 Platform BKMs

The following recommendations for platform reworks enable the supported I/O drivers (GPIO, I2C*, SPI, and HSUART) on Intel CRBs. This is not an exhaustive list of platform reworks.

4.1 Rework Bakersport Fab B USB3.0 Port

By default, the Bakersport Fab B CRB has an issue with the USB 3.0 port. This port fails to read several types of USB 3.0 drives and could not achieve USB 3.0 performance, as described in Table 4.

Note: Patriot Memory* 64 GB and EDGE Memory* DiskGO* 32 GB drives are not recommended to be used in EHCI mode.

Table 4. Rework Instructions for Bakersport Fab B USB 3.0 Port

Affected Platform	Bakersport boards (PBA# G72250-200 Rev 02) (Fab B)
Rework Steps	1. Un-stuff choke on L8A2. 2. Stuff R8A4 and R8A3 (0 ohms).

4.2 Rework Bakersport Fab B I2C* Port 6

By default, Bakersport Fab B CRB has an issue with I2C* port 6. This port fails to read and write because of incorrect resistor connections, as described in Table 5.

Table 5. Rework Instructions for Fab B I2C Port 6

Affected Platform	Bakersport boards (PBA# G72250-200 Rev 02) (Fab B)
Rework Steps	1. Un-stuff R5H9, R5H12, R5H8, R5H10. 2. Stuff R5H4 (22 ohms). 3. Stuff R5H3 (22 ohms).

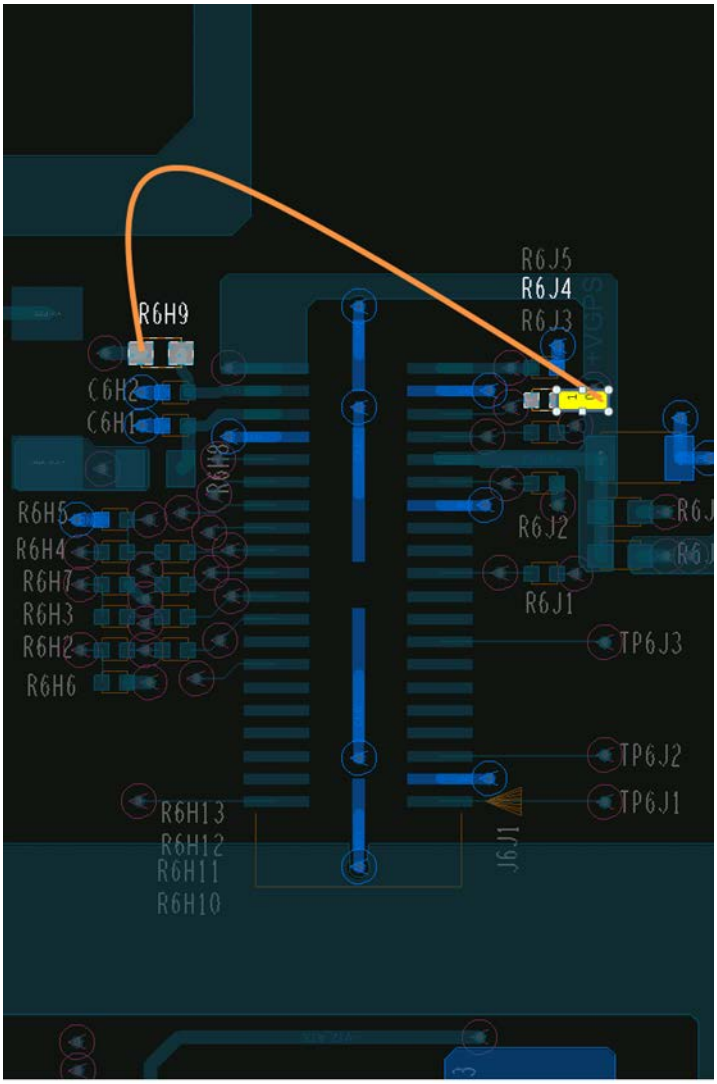
4.3 Rework UART in Bakersport and Bayley Bay

By default, Bakersport Fab B CRB and Bayley Bay Fab 03 CRB have an issue with the UART2 port. This port triggers an unwanted interrupt. Add a 10 K resistor to mitigate this issue, as described in Table 6. Figure 4 shows the rework layout. The 10 K PU resistor (denoted by a yellow box) is wired (denoted by an orange curved line) to R6H9.

Table 6. Rework Instructions for UART in Bakersport and Bayley Bay

Affected Platforms	Bakersport CRB (PBA# G72250-200 Rev 02) (Fab B) Bayley Bay Fab 3 CRB (IOTG configured) only
Rework Step	Place a 10 K resistor followed by a 28 AWG wire from R6J4 to R6H9.

Figure 6. Bakersport CRB and Bayley Bay CRB UART Rework Layout



§

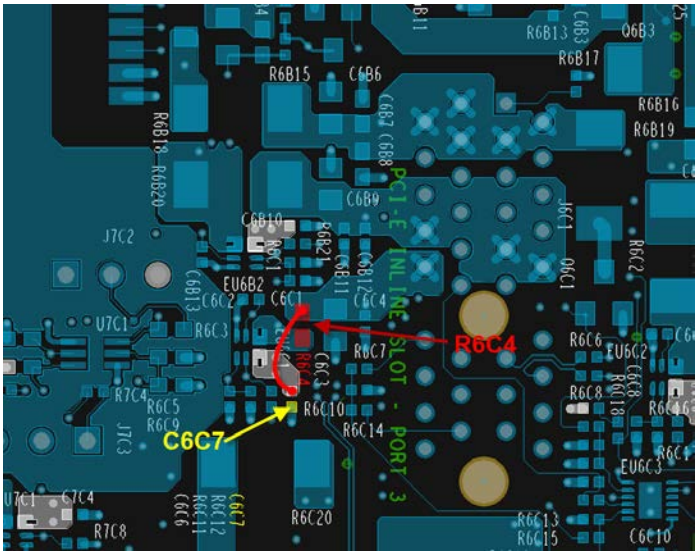
4.4 Rework Bayley Bay Fab 3 CRB PCIe* INLI Slot-Port 3

By default, Bayley Bay Fab 03 CRB has an issue with PCIe* Slot 3, as described in Table 7. This PCIe slot fails to detect a network card after shutdown followed by power up (without switching off the main power).

Table 7. Rework Instructions for Bayley Bay Fab 3 CRB PCIe* INLI Slot-Port 3

Affected Platform	Bayley Bay Fab 3 CRB (IOTG configured) only
Rework Steps	1. Remove R6C4. 2. Add a jumper wire from C6C7 to R6C4, as shown in Figure 7.
Reasons for Rework	NICs are not recognized in the Windows* operating system if the jumper block (J7C2) is configured to Desktop mode, pins [1–2]. Failure mode occurs in PCIe Slot3.

Figure 7. Bayley Bay Fab 3 CRB PCIe* Slot 3 Rework



4.5 Use Serial Port in Bayley Bay

The common serial port on the Bayley Bay CRB does not work. The actual serial port is the micro USB port near the COM port on the CRB board. Use a USB cable to connect the micro USB port on the CRB board to the USB port on the host machine (laptop or desktop).

Install a driver from the Future Technology Devices International (FTDI) Chip website (refer to Table 2) on the host machine to have the virtual COM port on the host machine communicate with the Bayley Bay CRB.

§

5.0 Software Driver BKMs

5.1 Disable the DMA Feature for I2C*

The seven I2C* controllers in the Intel Atom® E3800 processor use the Windows* registry to control the DMA feature:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\iaioi2c\Parameters]

"ForceDma"="0,0,0,0,0,0,0"
```

`ForceDma (string type)` consists of seven values mapped to the seven I2C controllers, which are device IDs from `0F41` to `0F47h`.

Value `0` disables the DMA, and I2C data will be read and written in PIO mode.

For values other than `0`, if data length is more than the specified value, I2C data will be read and written in DMA mode; if data length is less than the specified value, I2C data will be read and written in PIO mode.

By default, without any registry settings, I2C will use the PIO mode.

5.2 Set the Baud Rates of HSUART

This section describes the steps to set the Baud Rate of HSUART.

1. The baud rate is calculated based on the following method:

   ```
   Baud rate = (SourceClockFrequency) / (16 * divisor)
   Source Clock Frequency = 50000000 * PrescalerMValue
   /PrescalerNValue * 2
   For example, to set baud rate to 1M:
   Set PrescalerMValue = 64
   Set PrescalerNValue = 100
   SourceClockFrequency = 64,000,000
   ```

   The values of `SourceClockFrequency`, `PrescalerMValue`, and `PrescalerNValue` can be customized in the Windows* registry at:

   ```
   [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\iaiouart\Parameters]
   ```

   Reboot the system after setting the values.

2. The instance number from `0` to 3 should be combined with the above registry name for a different Host Controller. For example, to configure HSUART_1 to support baud rate 1M, 2M, and 4M, create and change the following registry settings:

```
"HSUartSourceClockFrequency_1"=dword:03D09000
"HSUartPrescalerMValue_1"=dword:000051ea
"HSUartPrescalerNValue_1"=dword:00007fff
```

3. To support baud rates between 300 and 115,200, change the registry settings for low speed source clock, M and N prescalers, as follows:

   ```
   "UartSourceClockFrequency"=dword:001c2000
   "UartPrescalerMValue"=dword:0000025a
   "UartPrescalerNValue"=dword:00007fff
   ```

   Refer to Section 27.2.3, Baud Rate Generator in the *Intel Atom® Processor E3800 Product Family Datasheet* (refer to Table 2).

§